

INHOLLAND

VHDL

# S88-n Protocol in VHDL

## *Authors:*

Koen Groot  
nummer

Ruben Pera  
551198

# Samenvatting

In dit onderzoeks verslag wordt de werking van het s88-n protocol onderzocht.  
Deze werking wordt geïmplementeerd met behulp van VHDL.

# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>3</b>
<b>2</b>	<b>Probleem Omschrijving</b>	<b>4</b>
<b>3</b>	<b>Specificaties</b>	<b>5</b>
<b>4</b>	<b>Vereisten</b>	<b>6</b>
<b>5</b>	<b>Analyse</b>	<b>7</b>
5.1	Hoe werkt het s88 protocol . . . . .	7
5.2	Hoe werkt een XILINX bord . . . . .	9
<b>6</b>	<b>Apendix</b>	<b>10</b>

# Hoofdstuk 1

## Inleiding

test2

# Hoofdstuk 2

## Probleem Omschrijving

De Hoofdvraag bij dit onderzoek luidt:

Hoe kan er op een XILINX bord met behulp van de hardware beschrijving taal VHDL een s88 testmachine geïmplementeerd worden?

Om de Hoofdvraag goed te kunnen beantwoorden zijn er de volgende deelvragen opgesteld:

1. Hoe werkt het s88 protocol?
2. Hoe werkt een XILINX Spartan 3E FPGA bord?
3. Hoe kan de hardware beschrijving taal VHDL gebruikt worden op het XILINX bord?
4. Hoe kan een schuifregister aangesloten worden op het XILINX bord?

Pas nadat alle Deelvragen beantwoordt zijn kan op de Hoofdvraag een volledig antwoordt geformuleerd worden.

# Hoofdstuk 3

## Specificaties

Onze specificaties staan hieronder genoteerd. Het is een checklist die ons gaat helpen bij het produceren van het eindeindelijke product.

1. De volgende Materialen gaan gebruikt worden:
  - Een XILINX Spartan 3E FPGA bord met per peripherals, zoals aansluiting voor stroom voorziening
  - Een schuifregister waarmee het s88 protocol getest kan worden
2. De volgende gereeschappen gaan gebruikt worden:
  - Een Computer waarop de VHDL code geschreven kan worden

# Hoofdstuk 4

## Vereisten

Waar het eindproduct aan moet voldoen:

- De code moet geschreven zijn in VHDL
- Het XILINX Spartan 3E FPGA bord moet kunnen communiceren met een schuifregister gebruik makend van het s88 protocol
- geen idee

# Hoofdstuk 5

## Analyse

### 5.1 Hoe werkt het s88 protocol

Het s88 wordt uitgelegd met behulp van onderstaande afbeelding.

Hierin is CENTRALE is de in dit project gebruikte XILINX bord.

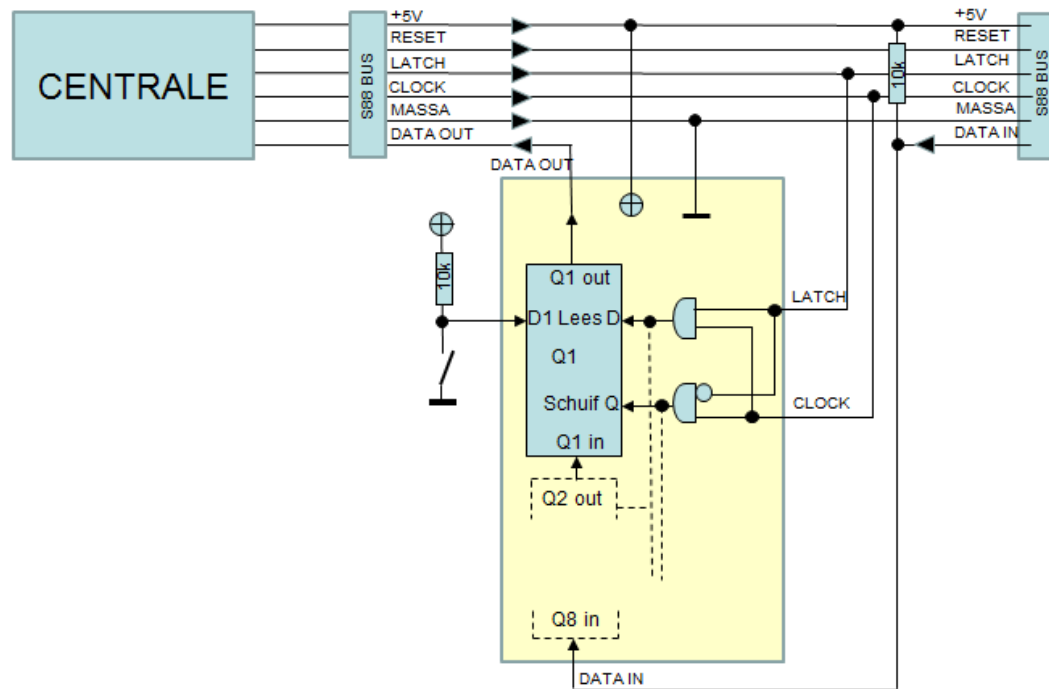
Een centrale, die hier CENTRALE heet, kan met behulp van het s88 protocol de rechter s88 bus uitlezen. Dit gebeurt met behulp van een schuifregister.

Met de DATA OUT poort van de CENTRALE s88 BUS wordt data ingelezen. Deze is verbonden met de Q1 out poort van het schuifregister, hiermee wordt dus informatie van het schuifregister naar de CENTRALE overgedragen.

De werking is als volgt, op het moment dat de CENTRALE een positieve flank geeft op de CLOCK en is de LATCH Hoog dan zal het schuifregister de data op Ingang D inlezen. Geeft de CENTRALE een positieve flank en is LATCH Laag dan zal de schuifregister één positie opschuiven.

Door een puls op RESET te geven worden de buffers gereset en zijn weer gereed voor het ontvangen van nieuwe data.





@onlineID, title = S88 TERUGMELDERS, date = 04-03-2016, url = <http://users.telenet.be/RedDe>

## **5.2 Hoe werkt een XILINX Spartan 3E FPGA bord**

# Hoofdstuk 6

## Appendix

```
1  Library ieee;
2  use ieee.std_logic_1164.all;
3  use IEEE.std_logic_unsigned.all;
4  use ieee.numeric_std.all;
5
6  --library UNISIM;
7  --use UNISIM.VComponents.all;
8  entity s88V2 is
9      port
10     (
11         OnboardClock, SW0, GPIO12 : in std_logic;
12         -- LED0, LED1, LED2, LED3, LED4, LED5, LED6,
13         LED7, GPIO14, GPIO16, GPIO17 : out std_logic;
14
15         clock          : in std_logic;
16         data_input     : in std_logic;
17         load_output    : out std_logic := '0';
18         clock_output   : out std_logic := '0';
19         reset_output   : out std_logic := '0';
20         data_output    : out std_logic := '0';
21         test_output    : out std_logic := '0'
22     );
23 end s88V2;
```

```

24
25
26 architecture s88TimingV2 of s88V2 is
27 signal CounterForTheClock    : integer := 0;
    -- gebruikt om de hoeveelheid klokslagen bij
    te houden
28 signal CounterForTheTime      : integer := 0;
    --
29 signal HertzToUse             : integer := 1;
30 begin
31
32     increase_CounterForTheClock_on_clock : process(
        clock)
33     begin
34         if rising_edge(clock) then
35             if CounterForTheClock = HertzToUse then
36                 --als er de gewenste hoeveelheid
                    klokslagen verstreken zijn
                    dan wordt de
                    CounterForTheTime verhoogd.
37                 if CounterForTheTime > 8 then
38                     CounterForTheTime <= 0;
39                 else
40                     CounterForTheTime <=
                        CounterForTheTime + 1;
41                 end if;
42                 CounterForTheClock <= 0;
43             else
44                 CounterForTheClock <= CounterForTheClock
                    + 1; --houdt bij hoeveel klokslagen
                    er zijn geweest,
45             end if;
46         end if;
47
48     end process increase_CounterForTheClock_on_clock
        ;
49
50     Read_IO_Case : process(CounterForTheTime)

```

```

51      begin
52          case CounterForTheTime is
53              when 0 =>
54                  --doe niks
55              when 1 =>
56                  load_output      <= '1';
57              when 2 =>
58                  clock_output     <= '1';
59              when 3 =>
60                  clock_output     <= '0';
61                  data_output <= '1';
62              when 4 =>
63                  reset_output     <= '1';
64              when 5 =>
65                  reset_output     <= '0';
66              when 6 =>
67                  load_output      <= '0';
68              when 7 =>
69                  clock_output     <= '1';
70              when 8 =>
71                  clock_output     <= '0';
72                  data_output <= '0';
73              when others =>
74                  --doe niks
75          end case;
76      end process Read_IO_Case;
77
78  end s88TimingV2;

```